



LAWRENCE  
LIVERMORE  
NATIONAL  
LABORATORY

# A Bayesian Approach to Multi-Sensor Track Correlation

M. Horsley

September 9, 2010

2010 AMOS Conference  
Maui, HI, United States  
September 14, 2010 through September 17, 2010

## **Disclaimer**

---

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

# **A Bayesian Approach To Multi-Sensor Track Correlation**

**Matthew A. Horsley**

*Lawrence Livermore National Laboratory*

LLNL-CONF-454076

## **ABSTRACT**

One of the primary goals of Space Situational Awareness is to locate objects in space and characterize their orbital parameters. This is typically performed using a single sensor. The use of multiple sensors offers the potential to improve system performance over what could be achieved by the use of a single sensor through increased visibility, increased accuracy, etc... However, to realize these improvements in practice, the association of data collected by the different sensors has to be performed in a reliable manner, with a quantifiable confidence level reported for each data association. Furthermore, sources of error such as track uncertainty and sensor bias have to be taken into account in order for the derived confidence levels to be valid. This paper will describe an approach to compute probabilities of association to support the integration of data collected by multiple sensors on a group of objects.

Multi-sensor data association is a fundamental problem in distributed multi-target multi-sensor tracking systems and involves finding the most probable association between object tracks. This is a challenging problem for a number of reasons. Each sensor may only observe a portion of the total number of objects, the object spacing may be small compared to a sensor's reported track accuracy, and each sensor may be biased. In addition, the problem space grows exponentially with the number of objects and sensors, making direct enumeration of the possible associations impractical for even modestly sized problems.

In this paper, the multi-sensor, multi-target likelihood function will be defined, with sensor bias included in the likelihood function. Sensor bias priors will be introduced and used to marginalize out the sensor bias. This marginalized likelihood will be incorporated into a Markov chain Monte Carlo data association framework and used to compute probabilities of association. In addition, the number of objects is treated as an unknown and probability distributions on this variable will also be produced. Simple problems involving modest numbers of targets and sensors will be described and the multi-sensor track correlation algorithm will be applied, with results presented and discussed.

## **1. INTRODUCTION**

This paper deals with the problem of reliably combining data collected by multiple sensors on a group of objects into a single, integrated picture. The first step is to assign the track files reported by the different sensors into groups representing objects being monitored by the sensor network. Thus each object in the complex being monitored will have at least one track report describing its orbital parameters, and potentially will have as many track reports as there are sensors reporting on the network. This last case represents a situation where information collected on the object by the different sensors in the network can be combined to improve system performance. In order to fully realize this improved system performance, the association function has to be performed in a way that will allow the system operator to determine the reliability of the result. This can be done by performing the multi-sensor data association within a probabilistic framework. The probability of association can then be used as a quantifiable confidence level and will enable the system operator to act with confidence on the resulting data product.

The problem of assigning track reports to objects being monitored is difficult for a number of reasons and can be divided into conceptual and algorithmic. On the conceptual side, track assignment can be considered a pattern matching problem, and the kinds of patterns being matched are noisy, oftentimes are incomplete, and can have a translational degree of freedom that will have to be considered in the solution. For example, each sensor may observe only a small portion of the total number of objects in the complex, the object spacing may be small compared to a sensor's reported track accuracy, and biased state estimates can be reported by a sensor. Sensor bias is a particularly challenging aspect of the problem. To help illustrate the nature of the bias, an example will be given comparing track reports generated by an optical telescope and a radar. When tracking objects with an optical sensor,

each observation contains data on the object being tracked, as well as a portion of the star field. This allows the object's orbital parameters to be estimated in a reference system defined by the star field. The optical tracking case can be compared to a radar tracking the same object. In this case the radar does not observe any stars and thus cannot produce an estimate of the objects orbital parameters in the same reference system as the optical sensor. However, the radar operator can perform a registration function in order to provide itself with an inertial reference system. The accuracy of an object's reported state estimate will be affected by any errors in the registration process. Assuming the radar has Global Positioning Satellite (GPS) data available, there should be enough information to adequately localize its position, and the dominant registration error is in the pointing error of the antenna. These pointing errors are typically referred to as bias and represent an offset in pointing angle from truth. Bias error will be present in all measurements made by the radar, and thus all objects tracked by the radar will share a common bias error, in addition to the random error associated with noise. Thus objects reported on by a radar can be expected to have random as well as systematic errors.

Developing an algorithmic solution is considered difficult primarily for two main reasons. The first is due to the fact that the bias is a continuous variable, while the assignments are discrete. This is an example of a mixed integer problem, which can be difficult to solve. The approach taken in this paper will develop an assignment likelihood that includes sensor bias, and then demonstrate how the sensor bias can be marginalized out, rendering the problem discrete. The second algorithmic difficulty is due to the large number of assignments that are possible. Problems of even modest size, say 11 objects being tracked by 4 sensors, result in a number of possible assignments that is greater than the number of stars in the observable universe,  $\sim 3 - 7 \times 10^{22}$ . This large number of possible assignments renders brute force enumeration approaches infeasible even for modest sized problems. Instead, this paper will describe a sampling technique based on a Markov Chain Monte Carlo framework that will allow a statistically rigorous and efficient way to sample from the space of possible assignments.

## 2. RELATED WORK

The data association problem and its application to track correlation have been studied extensively in the literature (e.g. [1] - [7], [9] - [11]). In [4], the problem of associating measurements to tracks in a probabilistic manner has been addressed using a Markov Chain Monte Carlo (MCMC) data association technique. This paper treated the data association problem in a Bayesian manner and used the MCMC technique to efficiently compute the probability of association. However, the issue of sensor bias was not addressed. In [5], [6] and [7] the issue of computing track-to-track assignments while explicitly accounting for sensor bias was addressed. The foundation for a rigorous approach to solving the problem while explicitly handling the issue of sensor bias was discussed in [7]. This approach relied on the fact that combinations of products and integrals of Gaussians may be reduced to matrix manipulations. However, the approach discussed in [7] involved the inversion of matrices that become very large for moderately sized problems, with no obvious way to simplify the problem. Also, computing probabilistic solutions to the track-to-track assignment problem invariably involves the examination of a large number of assignments. Likelihoods that involve many terms that are computationally expensive become prohibitive in problems with even a modest number of targets and/or sensors.

The approach taken in this paper makes use of the fact that the product of two Gaussians can be expressed as another Gaussian, see [8]. Using this fact, it will be shown that it is possible to formulate a multisensor-multitarget data association likelihood that explicitly accounts for sensor bias. The effect of sensor bias will be marginalized out, leaving a marginalized likelihood for use within an MCMC data association framework. In cases where there are many sensors and targets, it will be shown how the full likelihood can be approximated as a truncated product of Gaussians in order to allow solutions to be computed in an efficient manner.

## 3. PROBLEM FORMULATION

The formulation for the multisensor, multitarget assignment problem has been described in a number of different places, see [7],[9]-[12], and [13] for examples. The problem formulation as given in [13] is an excellent review of the problem. Let  $S$  denote a set of  $N_s > 1$  sensors. Each sensor reports on a number  $m_s$  of objects it is observing. Each track report will contain a (potentially biased) state estimate and error covariance matrix. It will be assumed that each object report refers to a single object, and thus no split tracks or merged tracks are present in the data sets reported by the sensors. The set of track reports for each sensor will be denoted  $X^s = \{\bar{x}_i^s, p_i^s\}$ ,  $i = 1, \dots, m_s$ .

Furthermore, each sensor will also have a bias,  $\vec{\beta}^s$  and a bias covariance matrix,  $P_\beta^s$ . The dimensionality,  $d$ , of the bias vector  $\vec{\beta}^s$  is the same as the state estimates,  $\vec{x}_i^s$ . The bias vector  $\vec{\beta}^s$  is the residual bias left over after the registration process has been performed by the radar and is not known. It will be incorporated into a multi-sensor, multi-target assignment likelihood and then marginalized, so its value is not needed. However, the sensor must have an estimate of the bias covariance matrix,  $P_\beta^s$ . This can be estimated using historical data available to the radar, or estimated by the user.

The object, then, is to generate probabilities of association for any given grouping of track reports. This requires a likelihood for a given association of track reports. Formulating a likelihood for a multi-sensor, multi-target assignment has been discussed in the literature (e.g. [7], [9] - [13]). The starting point for this formulation will use the likelihood defined in [12] for a set of  $N$  track reports assigned to a single object, whose state is represented by  $\tau$ :

$$p(x_i^1, \dots, x_i^{N_s} | \tau) = \frac{1}{(2\pi)^{\frac{dN_s}{2}}} \frac{1}{\prod_{s=1}^{N_s} |p_i^s|^{\frac{1}{2}}} \prod_{s=1}^{N_s} \exp\left(-\frac{1}{2}(x_i^s - \tau)^T (p_i^s)^{-1} (x_i^s - \tau)\right) \quad (3.1)$$

Each sensor can contribute at most one track report in the above formulation, which is the likelihood that the track reports from the different sensors all represent the same physical object. Using this as a starting point, we will now introduce sensor bias into the problem. Treating the bias  $\beta$  as a simple translational offset, we get:

$$p(x_i^1, \dots, x_i^{N_s} | \tau, \beta) = \frac{1}{(2\pi)^{\frac{dN_s}{2}}} \frac{1}{\prod_{s=1}^{N_s} |p_i^s|^{\frac{1}{2}}} \prod_{s=1}^{N_s} \exp\left(-\frac{1}{2}((x_i^s - \beta^s) - \tau)^T (p_i^s)^{-1} ((x_i^s - \beta^s) - \tau)\right) \quad (3.2)$$

If a particular sensor does not contribute a track report to the likelihood, we can incorporate a probability,  $P_{FA}^s$ , that the  $s^{th}$  radar did not see the  $i^{th}$  object into the likelihood in the following fashion:

$$p(x_i^1, \dots, x_i^{N_s} | \tau, \beta) = \prod_{s=1}^{\{\bar{N}_s\}} P_{FA}^s \frac{1}{(2\pi)^{\frac{dN_s}{2}}} \frac{1}{\prod_{s=1}^{\bar{N}_s} |p_i^s|^{\frac{1}{2}}} \prod_{s=1}^{\{\bar{N}_s\}} \exp\left(-\frac{1}{2}((x_i^s - \beta^s) - \tau)^T (p_i^s)^{-1} ((x_i^s - \beta^s) - \tau)\right) \quad (3.3)$$

where  $\{\bar{N}_s\}$  is the set of radars that do not contribute a track report to the likelihood and  $\{\bar{N}_s\}$  is the set of radars that do contribute a track report.

At this point, we are still dealing with the likelihood that the track reports from the different sensors all represent a single object. If multiple objects are being monitored, it is necessary to introduce a joint assignment variable,  $\omega$ . For

example, if 3 sensors are tracking 4 objects, a joint assignment might look like:  $\omega = \left\{ \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \\ 3 \end{pmatrix}, \begin{pmatrix} - \\ - \\ 4 \end{pmatrix} \right\}$ .

Each object in the joint assignment is represented by a grouping of track IDs, with the restriction that each sensor can contribute at most one track ID to any particular grouping. The first object in the joint assignment example shown above has 3 track reports: track ID 1 from sensor 1, track ID 1 from sensor 2, and track ID 2 from sensor 3. A dash is used to indicate no track report from that particular sensor is associated to that particular object in the joint assignment. The above example is just a single joint assignment out of a great number of possible joint assignments. The likelihood,  $\mathcal{L}(\omega)$ , of a joint assignment is simply the product of the individual object likelihoods,

$$\mathcal{L}(\omega) = \iint P^0(\beta) P^0(\tau) \prod_{i=1}^{|\omega|} p(x_{\omega(i)}^1, \dots, x_{\omega(i)}^{N_s} | \tau, \beta) d\beta d\tau \quad (3.4)$$

where we have introduced Gaussian priors on the unknown sensor biases,  $\beta$  and the unknown true object states,  $\tau$ . Finally, in order to compute the probability of any particular grouping of track reports into a cluster representing a single object being monitored by the network, it is necessary to compute the likelihood sum taken over all of the joint assignments that contain that particular grouping (denoted as the set  $\{\omega_{\{\alpha, \beta, \dots, \zeta\}}\}$ ), and normalize it by the summed likelihood computed using all possible assignments,

$$P_{\{\alpha, \beta, \dots, \zeta\}} = \frac{1}{Z} \sum_{i=1}^{|\omega_{\{\alpha, \beta, \dots, \zeta\}}|} \mathcal{L}(\omega) \quad (3.5)$$

where  $Z = \sum \mathcal{L}(\omega)$  is the normalizing factor and is the sum of likelihoods taken over all possible joint assignments.

### 3.1 LIKELIHOOD

At this point, the joint likelihood contains a number of unknowns that must be marginalized in order for the likelihood to be useful. In [8], a useful result is derived for performing integration on a product of Gaussians. A brief review of the result will be given in order to make the derivation of the likelihood clear, see [8] for more details. Following the derivation given in [8], we define a Gaussian density as

$$N_x(\mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right) \quad (3.6)$$

where  $\mu$  is the mean,  $\Sigma$  is the covariance, and  $x$  is a variable we want to integrate out. It is then shown that

$$\prod_{k=1}^K N_x(\mu_k, \Sigma_k) = \tilde{z} \cdot N_x(\tilde{\mu}, \tilde{\Sigma}) \quad (3.7)$$

where  $\tilde{\Sigma} = (\sum_{k=1}^K \Sigma^{-1})^{-1}$  and  $\tilde{\mu} = (\sum_{k=1}^K \Sigma_k^{-1})^{-1} (\sum_{k=1}^K \Sigma_k^{-1} \mu_k)$ , and

$$\tilde{z} = \frac{|2\pi\Sigma_d|^{-\frac{1}{2}}}{\prod_{k=1}^K |2\pi\Sigma_k|^{-\frac{1}{2}}} \prod_{i < j} \exp\left(-\frac{1}{2}(\mu_i - \mu_j)^T B_{ij}(\mu_i - \mu_j)\right) \quad (3.8)$$

where  $B_{ij} = \Sigma_i^{-1} (\sum_{k=1}^K \Sigma_k^{-1})^{-1} \Sigma_j^{-1}$ . Finally, using  $\int N_x(\tilde{\mu}, \tilde{\Sigma}) dx = 1$ , it is shown in [8] how to perform the integral on a product of K Gaussians, each with its own mean and covariance, but sharing a common offset  $x$ , to yield the following expression

$$\int_{\mathbb{R}^d} \prod_{k=1}^K N_x(\mu_k, \Sigma_k) dx = \tilde{z} \quad (3.9).$$

This result will be used to marginalize out the unknown truth state of each object in the assignment hypothesis as well as the unknown bias from each sensor. However, rather than starting with the general case, we will first derive the marginalized likelihood for the special case of 2 sensors, which see all of the objects. After that, it will be generalized for N sensors.

### 3.2 SPECIAL CASE: 2 SENSORS

In the case of 2 sensors, the truth state drops out when formulating the likelihood, and the bias of the two sensors can be expressed as a single relative bias,  $\beta_{rel} = \beta^1 - \beta^2$ , resulting in the following expression for the likelihood of a single object assignment,

$$p^{2sen}(x_i^1, x_j^2 | \beta_{rel}) = \frac{1}{\sqrt{|2\pi(p_i^1 + p_j^2)|}} \exp\left(-\frac{1}{2}(x_i^1 - x_j^2 - \beta_{rel})^T (p_i^1 + p_j^2)^{-1} (x_i^1 - x_j^2 - \beta_{rel})\right).$$

The likelihood for a joint assignment is then

$$\mathcal{L}^{2sen}(\omega) = \int \mathcal{N}_{\beta_{rel}}(\bar{\beta}_{rel}, P_{rel.bias}) \prod_{i=1}^{|\omega|} p^{2sen}(x_{\omega(i)}^1, x_{\omega(i)}^2 | \beta_{rel}) d\beta_{rel}.$$

Define the prior on the relative bias term,  $\mathcal{N}_{\beta_{rel}}(\bar{\beta}_{rel}, P_{rel.bias})$ , as a Gaussian centered on  $\bar{\beta}_{rel}$  with a covariance  $P_{rel.bias}$ . Assuming that the two radars have performed registration, we can take the  $\bar{\beta}_{rel}$  term to be zero. In order to simplify notation, denote the difference in states on the  $i^{th}$  pairing in the joint assignment as  $\Delta x_{\omega(i)}^{12} = x_{\omega(i)}^1 - x_{\omega(i)}^2$  and let  $p_{\omega(i)}^{12} = p_{\omega(i)}^1 + p_{\omega(i)}^2$ . Using (3.8), the marginalized likelihood of a joint assignment for 2 sensors is

$$\mathcal{L}^{2sen}(\omega) = \frac{|2\pi\Sigma_d|^{-\frac{1}{2}}}{|2\pi P_{rel.bias}|^{-\frac{1}{2}} \prod_{i=1}^{|\omega|} |2\pi(p_{\omega(i)}^{12})|^{-\frac{1}{2}}} \prod_{i=1}^{|\omega|} \exp\left(-\frac{1}{2}(\bar{\beta}_{rel} - \Delta x_{\omega(i)}^{12})^T B_{0i}(\bar{\beta}_{rel} - \Delta x_{\omega(i)}^{12})\right) \prod_{k < l} \exp\left(-\frac{1}{2}(\Delta x_{\omega(k)}^{12} - \Delta x_{\omega(l)}^{12})^T B_{kl}(\Delta x_{\omega(k)}^{12} - \Delta x_{\omega(l)}^{12})\right), \quad (3.10)$$

where  $B_{0i} = P_{rel.bias}^{-1} \Sigma_d (p_{\omega(i)}^{12})^{-1}$ ,  $B_{kl} = (p_{\omega(k)}^{12})^{-1} \Sigma_d (p_{\omega(l)}^{12})^{-1}$ , and  $\Sigma_d = (P_{rel.bias}^{-1} + \sum_{k=1}^{|\omega|} (p_{\omega(k)}^{12})^{-1})^{-1}$ .

Examining the equation for  $\mathcal{L}^{2sen}(\omega)$ , we have two products of Gaussians. The first product is taken over each pairing in the joint assignment. Intuitively, what this is showing us is that a ‘good’ likelihood will have differences between track reports that will be on the order of the combined uncertainty for the two track reports plus the uncertainty in the bias. The second product is more interesting, as it involves comparisons of track state differences between the various objects in the assignment. Intuitively, what this is telling us is that a ‘good’ assignment is one where all of the pairings in the joint assignment have similar state differences.

In practice, computing probabilities of association involves calculating the likelihoods of a great number of joint assignments. Every term in the likelihood involves  $\Sigma_d$ , which can vary depending on the assignment. Thus, every term in the likelihood will have to be re-computed (except for the determinant’s in the denominator). However, if we make the following approximations;

$$B_{0i} \rightarrow \tilde{B}_{0i} = (P_{\text{rel.bias}} + p_{\omega(i)}^{12})^{-1} \quad \text{and} \quad B_{kl} \rightarrow \tilde{B}_{kl} = (p_{\omega(k)}^{12} + p_{\omega(l)}^{12})^{-1},$$

every term in the likelihood (except for the determinant in the numerator) is independent of  $\Sigma_d$ . The error in doing this is not great, and the benefit is that the likelihood is now a function of a number of variables that can be computed once and stored in a lookup table.

### 3.3 GENERAL CASE: N SENSORS

In the general case involving more than 2 sensors, we can no longer ignore  $\tau$ . First, in order to perform the marginalization of the truth state of the  $k^{\text{th}}$  object we introduce a Gaussian prior,

$$Pr = \frac{1}{(2\pi)^{\frac{d}{2}} |p_\tau|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\bar{\tau} - \tau)^T (p_\tau)^{-1} (\bar{\tau} - \tau)\right) = N_\tau(\bar{\tau}, p_\tau) \quad (3.11)$$

into the likelihood expression (3.3).  $\bar{\tau}$  is the prior expectation on the object's state and  $p_\tau$  is the covariance matrix. This will result in a product of Gaussians that all share a common offset,  $\tau$ . Using the results of (3.8), we get the expression for the marginalized likelihood for a single object,  $k$ :

$$\begin{aligned} \mathcal{L}_k = & \int d\tau^k N_\tau(\bar{\tau}^k, p_\tau^k) \prod_{s=1}^{N_s} N_\tau(x_k^s - \beta^s, p_k^s) = \\ & \frac{(2\pi)^{\frac{d}{2}} |\Sigma_d|^{\frac{1}{2}}}{(2\pi)^{\frac{d}{2}} |p_\tau^k|^{\frac{1}{2}} \prod_{s=1}^{N_s} (2\pi)^{\frac{d}{2}} |p_k^s|^{\frac{1}{2}}} \prod_{i=1}^{N_s} \exp\left(\bar{\tau}^k - (x_k^i - \beta^i), (p_\tau^k)^{-1} \Sigma_d (p_k^i)^{-1}\right) \prod_{i < j}^{N_s} \exp\left(\Delta x_k^{ij} - \Delta \beta^{ij}, (p_k^i)^{-1} \Sigma_d (p_k^j)^{-1}\right) \end{aligned}$$

where  $\exp(x, P) = \exp(-\frac{1}{2}x^T P x)$  and we have ignored the case where a sensor does not contribute a track report in order to reduce the length of the resulting expressions. Similar expressions can be obtained for the cases where a subset of the sensors do not contribute a track report. The integral contains three main terms: a factor consisting of matrix determinants, and two products of Gaussians. The first product is a comparison of the track reports from each sensor against the prior expectation for the object state. The second set of terms is a pair-wise comparison of track states over all possible sensor pairings for that particular object.

The likelihood for the joint assignment is simply the product of the likelihoods for each object in the joint assignment,  $\mathcal{L}(\omega) = \prod_k \mathcal{L}_k$ . In order to marginalize the unknown bias states, we proceed in the same manner as

above, introducing bias priors  $Pr_{\beta^i} = \frac{1}{2\pi^{\frac{d}{2}} |p_{\beta^i}|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\bar{\beta}^i - \beta^i)^T P_{\beta^i}^{-1} (\bar{\beta}^i - \beta^i)\right)$  for each of the radars, where

the index  $i = 1, \dots, N_s$ . Typically, the radar will have performed registration, so the  $\bar{\beta}$  terms can be taken to be zero. Then we perform the integration as before, using the results of (3.8). We start by gathering all terms that contain  $\beta^1$  and perform the marginalization. After performing the integration over the first bias state, we will gather all the terms that contains  $\beta^2$  and again apply (3.8), continuing until we have marginalized all bias states.

Unfortunately, the number of terms grows substantially when performing the integration and the calculation becomes exceptionally tedious and results in an expression that is too long to be included in this paper. Interested parties can contact the author for a handwritten set of notes where the special case of 3 sensors has been derived. The expression contains products such as:

$$\begin{aligned} & \prod_{i=1}^{|\omega|} \exp\left(\bar{\beta}^1 - \Delta x_{\omega(i)}^{01}, P_{\beta^1}^{-1} \tilde{\Sigma}_d (P_\tau^{-1} \Sigma_d P_{\omega(i)}^1)^{-1}\right) \\ & \prod_{i=1}^{|\omega|} \exp\left(\bar{\beta}^2 - \Delta x_{\omega(i)}^{02}, P_{\beta^2}^{-1} \tilde{\Sigma}_d (P_\tau^{-1} \Sigma_d P_{\omega(i)}^2)^{-1}\right) \\ & \prod_{i=1}^{|\omega|} \exp\left(\bar{\beta}^3 - \Delta x_{\omega(i)}^{03}, P_{\beta^3}^{-1} \tilde{\Sigma}_d (P_\tau^{-1} \Sigma_d P_{\omega(i)}^3)^{-1}\right) \end{aligned}$$

and

$$\begin{aligned} & \prod_{i < j}^{|\omega|_{12}} \exp \left( \Delta x_{\omega(i)}^{12} - \Delta x_{\omega(j)}^{12}, \left( (P_{\omega(i)}^1)^{-1} \Sigma_d (P_{\omega(j)}^2)^{-1} \right)^{-1} \Sigma'_d \left( (P_{\omega(i)}^1)^{-1} \Sigma_d (P_{\omega(j)}^2)^{-1} \right)^{-1} \right) \\ & \prod_{i < j}^{|\omega|_{13}} \exp \left( \Delta x_{\omega(i)}^{13} - \Delta x_{\omega(j)}^{13}, \left( (P_{\omega(i)}^1)^{-1} \Sigma_d (P_{\omega(j)}^3)^{-1} \right)^{-1} \Sigma'_d \left( (P_{\omega(i)}^1)^{-1} \Sigma_d (P_{\omega(j)}^3)^{-1} \right)^{-1} \right) \\ & \prod_{i < j}^{|\omega|_{23}} \exp \left( \Delta x_{\omega(i)}^{23} - \Delta x_{\omega(j)}^{23}, \left( (P_{\omega(i)}^2)^{-1} \Sigma_d (P_{\omega(j)}^3)^{-1} \right)^{-1} \Sigma'_d \left( (P_{\omega(i)}^2)^{-1} \Sigma_d (P_{\omega(j)}^3)^{-1} \right)^{-1} \right) \end{aligned}$$

as well as a great many more such terms. In the above expressions,  $\Sigma_d$  is the covariance term resulting from the marginalization of the truth on the  $i^{\text{th}}$  object, while  $\Sigma'_d$  and  $\Sigma''_d$  refer to the marginalization covariance matrices obtained from the marginalization of the first two bias terms. The symbol  $\prod_{i=1}^{|\omega|_{12}}$  refers to a product taken over all objects in the joint assignment that contain track reports from the first and second sensor, and in general for any sensor pairing  $\alpha, \beta$  we have  $\prod_{i=1}^{|\omega|_{\alpha\beta}}$ .

The other terms appearing in the above equations are defined as  $\Delta x_{\omega(i)}^{0s} = \tau_{\omega(i)} - x_{\omega(i)}^s$  and  $\Delta \bar{\beta}^{ij} = \bar{\beta}^i - \bar{\beta}^j$ . In addition to the large number of terms, the complexity of each term is substantial as it involves the multiplication and inversion of a great many matrices, which can be computationally expensive. So, similar to what was done for the special case of two sensors, a simplification will be made to render the marginalized likelihood for the general case of  $N_s$  sensors useful for computations,

$$\begin{aligned} \mathcal{L}_{\text{marg}} = \prod P_{FA} \frac{(2\pi)^{\frac{d}{2}} |\Sigma_d|^{\frac{1}{2}}}{\prod_{i=1}^{N_s} (2\pi)^{\frac{d}{2}} |P_{\beta^i}|^{\frac{1}{2}} \prod_{i=1}^{N_s} \prod_{k=1}^{|\omega|} (2\pi)^{\frac{d}{2}} |P_{\omega(k)}^i|^{\frac{1}{2}}} & \prod_{i < j}^{N_s} \left( \prod_{k=1}^{|\omega|} \exp \left( \Delta x_{\omega(k)}^{ij}, \left( P_{\omega(k)}^{ij} + P_{\beta^i} + P_{\beta^j} \right)^{-1} \right) \right) \\ & \prod_{i < j}^{N_s} \left( \prod_{k < l}^{|\omega|} \exp \left( \Delta x_{\omega(k)}^{ij} - \Delta x_{\omega(l)}^{ij}, \left( P_{\omega(k)}^{ij} + P_{\omega(l)}^{ij} \right)^{-1} \right) \right), \end{aligned} \quad (3.12)$$

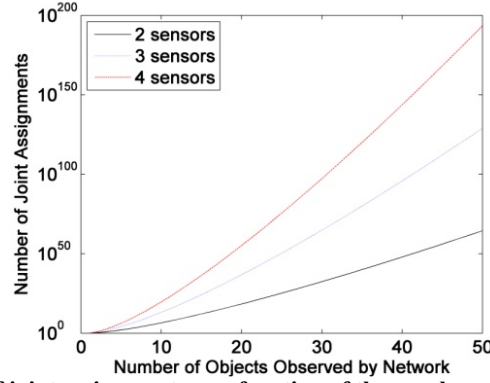
where  $\tilde{\Sigma}_d^{-1} = \sum_{i=1}^{N_s} P_{\beta^i}^{-1} + \sum_{k=1}^{|\omega|} \sum_{i=1}^{N_s} (P_{\omega(k)}^i)^{-1}$ .

In addition to this simplified likelihood being substantially smaller in the number of terms contained in the expression, each term can be pre-computed and stored in a lookup table. The likelihood itself contains two double products of Gaussians. The first is a double product taken over all sensor pairings for all object assignments. The covariance term includes the bias covariance matrices and thus favors track pairings that are close relative to the combined uncertainty of the track states and bias. The second double product involves comparisons in the differences between the track states contained in the different assignments. Thus, this favors joint assignments where the state differences between every pairing are all consistent with one another.

#### 4. MARKOV CHAIN MONTE CARLO

Markov Chain Monte Carlo (MCMC) represents a general class of computing techniques that are used in a great many different fields, such as physics, chemistry, biology, and computer science. The method has been extensively discussed in the literature and has a sound theoretical foundation. For a review of the topic, many excellent tutorials exist, see [19] for an example. The goal is to calculate the probability of association for a given grouping of track reports generated by a sensor network using equations (3.5) and (3.12). The expression in the denominator of (3.5) involves a summation over all possible global assignments. The direct enumeration of all the assignments is only possible in cases involving a very modest number of track reports and contributing sensors, see fig. 1. However, an approximation can be computed very easily and efficiently using the MCMC method. The exact approach will be discussed below, but a very brief summary of the method is given here. The MCMC technique is used to generate a large number of joint assignments. Each joint assignment generated by the algorithm is sampled in direct proportion to its probability. Thus joint assignments that are very likely will be sampled many times by the algorithm. Joint assignments that are not very likely will only be sampled a few times (or not at all) by the algorithm. Once a set of joint assignments has been generated, any sort of statistical calculation can be performed, such as computing a mean, variance or credible interval. Accuracy can be improved simply by sampling a greater number of joint assignments. The importance of an efficient method for computing the joint assignment likelihood lies in the fact that the MCMC method will typically require a large number of iterations. Eqn. (3.12) is the product of a number of terms, each of which can be computed ahead of time and stored in a lookup table. Thus the calculation consists entirely of indexing a lookup table and can be performed efficiently.





**Figure 1** Growth in the number of joint assignments as a function of the number of objects observed by network.

Following the explanation given in [14], the method starts by defining a Markov Chain,  $\mathcal{M}$ , which will be used to generate samples from a probability distribution  $\pi$  on the space of all possible assignments,  $\Omega$ . By construction,  $\mathcal{M}$  will have states  $\omega \in \Omega$  and a stationary distribution  $\pi(\omega)$ . If the current state of the chain is  $\omega$ , a new state  $\omega'$  will

#### **MCMC Algorithm**

sample  $U$  from uniform  $[0,1]$

**if**  $U < \frac{1}{2}$  **then**

    proposed joint assignment remains unchanged

**else**

    sample  $V$  from  $[1,2,3,4]$  uniformly at random

**if**  $V = 1$  **then** (split move)

    split a single track grouping into two separate groupings uniformly at random

**else if**  $V = 2$  **then** (merge move)

    merge two track groupings into a single object grouping uniformly at random

**else if**  $V = 3$  **then** (switch move)

    switch single track assignment from object grouping  $i$  to object grouping  $j$ , uniformly at random

**else if**  $V = 4$  **then** (swap move)

    swap track assignment  $\alpha$  from object assignment  $i$  with track assignment  $\beta$  from object grouping  $j$ , uniformly at random

**end if**

**end if**

    accept/reject proposed joint assignment using  $A(\omega, \omega')$

**Table 1** MCMC Algorithm.

be proposed for the next step of the chain. The new state will be generated following a proposal distribution,  $q(\omega, \omega')$ , which is dependent on the current state of the chain. The proposed move  $\omega'$  will be accepted with a probability

$$A(\omega, \omega') = \min \left( 1, \frac{\pi(\omega')q(\omega', \omega)}{\pi(\omega)q(\omega, \omega')} \right).$$

By the ergodic theorem, it can be shown that  $\mathcal{M}$  will converge to its stationary distribution as long as  $\mathcal{M}$  satisfies a few requirements, such as being irreducible and aperiodic (see [14] & [16] for details). Using a uniform proposal distribution (see table 1) allows a simplification in the acceptance probability,

$$A(\omega, \omega') = \min \left( 1, \frac{\pi(\omega')}{\pi(\omega)} \right).$$

The acceptance probability  $A(\omega, \omega')$  now depends only on the ratio of the likelihoods, and does not require knowledge of the normalizing constant  $Z$ . This is exactly what we need. It allows us to generate a large number of joint assignments in direct proportion to their probabilities, but without requiring an exhaustive enumeration of the states. The only requirement is to compute the likelihood of a joint assignment. The Markov Chain will be simulated using the algorithm described in table 1, which is loosely based on the chain discussed in [14]. Thus,

using the method described in [14] and [15], the probability of a particular object assignment for two sensors is estimated using

$$\hat{P}_{jk} = \frac{1}{n_{mc} - n_{bi}} \sum_{n=n_{bi}}^{n_{mc}} \mathbb{I}((j, k) \in \omega_n),$$

where  $\hat{P}_{jk}$  is the association probability of the  $j^{\text{th}}$  track from sensor 1 and the  $k^{\text{th}}$  track from sensor 2, and  $n_{mc}$  and  $n_{bi}$  are the total number of Monte Carlo iterations and an initial number of discarded moves performed in the calculation, respectively. Thus the assignment probability is obtained by simply counting the number of joint assignments that contain the  $(j, k)$  pair and normalizing by the total number of iterations used in the calculation. This can be directly extended to the case involving 3 or more sensors by counting the number of joint assignments that contain the grouping  $\{\alpha, \beta, \dots, \zeta\}$  and normalizing by the total number of iterations used in the calculation,

$$\hat{P}_{\{\alpha, \beta, \dots, \zeta\}} = \frac{1}{n_{mc} - n_{bi}} \sum_{n=n_{bi}}^{n_{mc}} \mathbb{I}(\{\alpha, \beta, \dots, \zeta\} \in \omega_n).$$

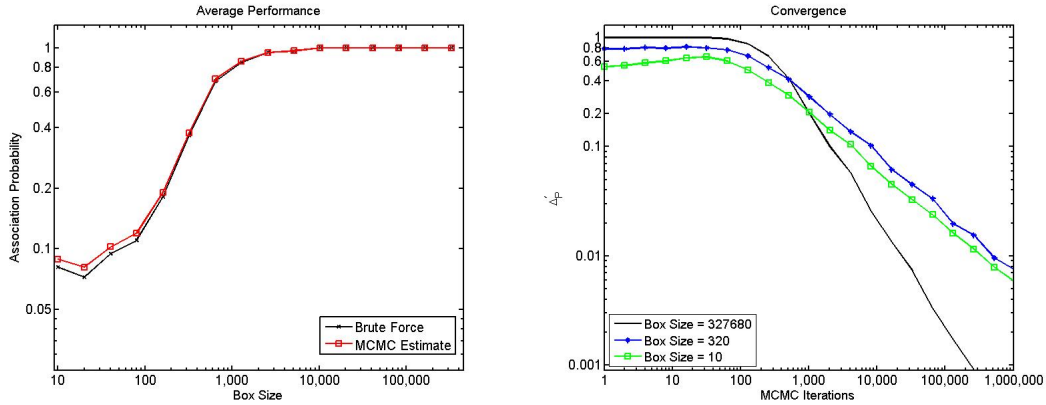
## 5. SIMULATION RESULTS

In this section, a simple simulation of a sensor network will be described and how the multisensor-multitarget track correlation algorithm performs using the simulated data as input. In [14], a metric was defined for measuring the performance of a data association algorithm,  $\Delta_P = \max_{j,k} |P_{jk} - \hat{P}_{jk}|$ .  $P_{jk}$  is the association probability of track report  $j$  from sensor 1 and track report  $k$  from sensor 2. In this paper,  $P_{jk}$  was obtained by direct enumeration of all possible joint assignments and applying (3.5) and (3.12), i.e. a brute force calculation of the true probability.  $\hat{P}_{jk}$  is the association probability estimated using the MCMC algorithm described above. Since the metric uses a brute force approach to compute the true probability, the performance can only be characterized for modest-sized problems. While the metric is directly applicable to the 2 sensor case, it can be directly extended to handle cases involving multiple sensors,  $\Delta'_P = \max_{j,k,\dots,n} |P_{jk\dots n} - \hat{P}_{jk\dots n}|$ .

A simple example involving 3 sensors tracking a group of 5 objects is used to generate the results shown in figure 2. Each sensor reports on a subset of the objects in the group. The subset reported by each sensor was chosen in a random fashion, where the number of objects observed by a sensor was allowed to vary from 2 up to a maximum of 5. A ‘truth’ scene was constructed by generating 5 object states, uniformly at random, contained within a d-dimensional box of variable length. In this example,  $d = 2$  even though the algorithm can handle an arbitrary number of dimensions. The length of a box side was allowed to vary from 10 to a maximum of 327,680 (the units are arbitrary). The truth scene was then used to simulate the observations reported by each sensor. For each sensor, a bias vector was created using a Gaussian distribution having a mean of zero and a d-dimensional spherical covariance with a radius defined by  $\sigma_{bias}^2 = 1000^2$ . The objects observed by each sensor had track reports simulated by applying a Gaussian noise model to the truth and adding the bias to each object reported by the sensor.

The covariance matrix for each object was kept the same for all objects and all sensors,  $p_i^s = \begin{pmatrix} 100^2 & 0 \\ 0 & 100^2 \end{pmatrix}$ . For each box size, 200 simulations were run and the results were stored for analysis. For each simulation, the assignment probabilities were computed by direct enumeration using (3.5). This is compared to the association probability estimated using the MCMC algorithm described in this paper. The results are shown in the left panel in figure 2. The MCMC results were computed by allowing the algorithm to run for 2 million iterations. In the right panel the rate of convergence is shown as a function of the number of iterations. The metric used to measure the convergence performance of the algorithm is  $\Delta'_P = \max_{j,k,\dots,n} |P_{jk\dots n} - \hat{P}_{jk\dots n}|$ . For cases where the box size is much larger than the combined uncertainty of the track states and sensor bias, the convergence is rapid. As the box size is made smaller, the convergence is not as rapid, but still shows good performance yielding solutions that are 0.5% from the true probability within 100,000 Monte Carlo iterations.

The number of iterations that the MCMC algorithm executes will determine the accuracy of the estimated assignment probabilities, i.e. the longer the algorithm runs, the closer the estimated probabilities will be to the true probability. In certain cases the mixing time of the chain can be derived (see [14] for excellent example) and used to bound the number of iterations required, but in many situations this is not possible and a convergence diagnostic must be used. There are several methods discussed in the literature (see [17] for a review of the topic). Gelman and Rubin [18] proposed a general method for monitoring the convergence of the MCMC algorithm. The method requires a number,  $N$ , of MCMC chains to be implemented in parallel. Convergence is diagnosed by comparing



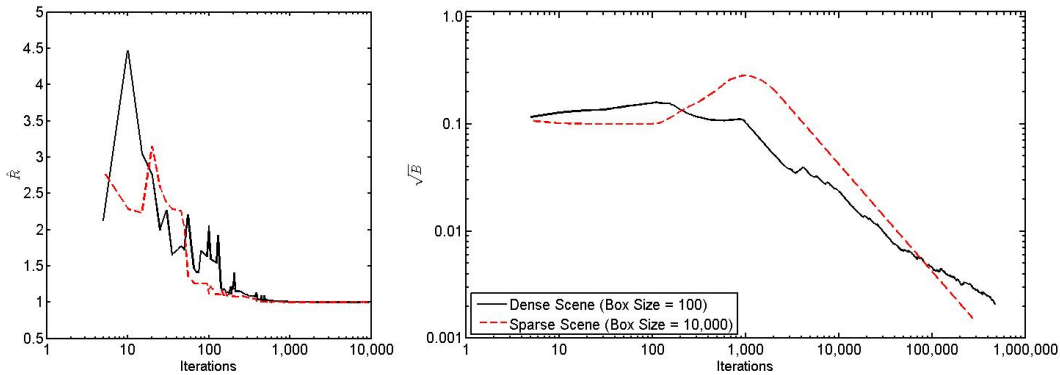
**Figure 2** The association probability as a function of scene density (left panel) and convergence as a function of the number of iterations performed (right panel).

the within-chain and between chain variances of the parameters being estimated by the MCMC algorithm (i.e. assignment probabilities). The variance of the parameter  $\theta$ , is defined as a weighted average of  $W$  and  $B$ ,

$$Var(\theta) = (n - 1) \frac{W}{n} + \frac{B}{n}$$

where  $n$  is the number of iterations performed,  $W$  is the mean of the empirical variance within each chain, and  $B/n$  is the empirical variance computed using all the chains. The main idea is that when the chains have converged, then both estimates will be unbiased. If the algorithm has not converged, the between chain variance will tend to overestimate the variance since the chains were started at different points and the within chain variance will tend to underestimate the variance since no chain will have had enough time to explore the entire space of solutions. A scale

reduction factor is defined as  $\hat{R} = \sqrt{Var(\theta)/W}$  and used to monitor convergence. Values of  $\hat{R} \lesssim 1.1$  are an indication that convergence has been achieved. Thus a procedure for computing probabilities of assignment can be done in the following manner.  $N$  chains are instantiated and allowed to run for a certain number of iterations. The scale reduction factor is monitored. When  $\hat{R}$  has fallen to a value of 1.1, the MCMC iterations accumulated up to then are discarded, and the chains are allowed to continue. At regular intervals, the between chain variance,  $B$ , can be computed for any assignment the user cares to keep track of. This will give an estimate of the uncertainty for the



**Figure 3** Example of Gelman-Rubin convergence diagnostic for two cases involving 15 objects in scene.

assignment probability being monitored. Once the uncertainty of the assignment has reached a user defined threshold, the algorithm can be stopped. Figure 3 shows the behaviors of the scale reduction factor and between chain variance for a particular assignment. The scale reduction factor and assignment probability variance both improve steadily as the number of iterations is increased.

## 6. CONCLUSIONS

The multi-sensor, multi-target track association likelihood was developed for cases involving sensor bias. The exact bias marginalized likelihood was derived for the special case of two sensors. A simple method of approximating the exact likelihood was described for enabling its use in a Markov Chain Monte Carlo program. For the general case of N sensors, a procedure was described for computing the exact bias marginalized likelihood. Because of the resulting complexity, an approximate likelihood for the multisensory case was derived and used in a MCMC algorithm. The MCMC algorithm results were compared to the exact results for the special case of 3 sensors. Performance was good in cases involving more than 100,000 MCMC iterations. A simple procedure was described for diagnosing convergence of the MCMC algorithm and showed good performance in dense and sparse scenes.

## 7. ACKNOWLEDGEMENTS

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

## 8. REFERENCES

1. Bar Shalom and Fortmann, *Tracking and Data Association*, Academic Press, New York, 1988.
2. S. Blackman, *Multiple-Target Tracking and Radar Applications*, Artech House, Dedham, MA, 1986.
3. S. Mori, K.C. Chang, and C.Y Chong, "Performance analysis of optimal data association with applications to multiple target tracking," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., 2, pp. 183-236, Artech House, Norwood, MA, 1992.
4. S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for General Multiple Target Tracking Problems," *Proc. 43rd IEEE Conf. Decision and Control*, 2004.
5. M. Levedahl, "An Explicit Pattern Matching Assignment Algorithm," Signal and Digital Processing of Small Targets (O.E. Drummond, Editor), Proc. of SPIE Vol.6 4728, 2002 .
6. Papageorgious, D.J., Sergi, J.-D., "Simultaneous Track-to-Track Assignment with Bias Removal using Multistart Local Search", *IEEE Aerospace Conference*, pp. 1-14, Mar. 2008
7. J. Ferry, "Exact Bias Removal for the Track-to-Track Association Problem", *12<sup>th</sup> International Conference on Information Fusion*, Seattle, WA, USA, July 6-9, 2009
8. P. Ahrendt, "The Multivariate Gaussian Probability Distribution", available online at website: [http://www2.imm.dtu.dk/pubdb/views/edoc\\_download.php/3312/pdf/imm3312.pdf](http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/3312/pdf/imm3312.pdf)
9. S. Deb, M. Yeddanapudi, K. Pattipati and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation", *IEEE Trans. On Aerospace and Electronic Systems*. 33(2):523-538, 1997
10. Y. Zhang, H. Leung, M. Blanchette, T. Lo and J. Litva. "An efficient decentralized multiradar multitarget tracker for air surveillance," *IEEE Trans. On Aerospace and Electronic Systems*. 33(4):1357-1363, 1997
11. A.J. Kanyuckh and R.A. Singer, "Correlation of multiple site track data," *IEEE Trans. On Aerospace and Electronic Systems*. 6:180-187, 1970
12. Kaplan, L.; Blair, W.; , "Assignment costs for multiple sensor track-to-track association," In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden. June 2004, 1231-1237.
13. Kaplan, L.; Bar-Shalom, Y.; Blair, W.; , "Assignment costs for multiple sensor track-to-track association," *Aerospace and Electronic Systems, IEEE Transactions on* , vol.44, no.2, pp.655-677, April 2008
14. S. Oh, S. Sastry, "A polynomial-time approximation algorithm for joint probabilistic data association," in Proc. of the American Control Conference, Portland, OR, June 2005.
15. W. Gilks, S. Richardson, D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series. Chapman and Hall, 1996.
16. G. Roberts,"Markov chain concepts related to sampling algorithms," in *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series. Chapman and Hall, 1996.
17. M. K. Cowles, B. P. Carlin, "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review," *Journal of the American Statistical Association*, Vol. 91, 1996
18. Gelman, A. and Rubin, D.B., "Inference from Iterative Simulation Using Multiple Sequences (with discussion)." *Statistical Science*, 7,457-511 1992
19. L. Tierney, "Markov Chains for Exploring Posterior Distributions", *The Annals of Statistics*, vol 22, no 4 1994